

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Aplicación de gestión de mascotas

OURPETS

Iván Zorita Bote

Tutor: Alfonso Ortega De La Puente

Julio 2016

Aplicación de gestión de mascotas OURPETS

AUTOR: Iván Zorita Bote

TUTOR: Alfonso Ortega De La Puente

Dpto. Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

JULIO de 2016

Resumen

En la actualidad existen un gran número de familias con animales en casa. El incremento de ese número de familias con animales ha ido de la mano con el incremento de la preocupación de esas familias, y de la sociedad en general, por la salud de los animales. Para ello es necesario llevar un control de las visitas al veterinario, vacunas, etc. Pero esto en ocasiones supone tener que guardar muchos papeles.

Esta situación ha sido la motivación para realizar este proyecto de fin de grado. Después de realizar el análisis del estado del arte se han definido los objetivos para el desarrollo de una aplicación nativa en Android, Ourpets, que es un sistema que permite gestionar la información relevante sobre el historial médico de las mascotas.

El desarrollo de la aplicación se ha realizado implementando tres módulos, que llevan a cabo tres funcionalidades distintas. El primero consiste en la aplicación móvil y la interfaz de usuario, la implementación de la aplicación se ha llevado a cabo mediante el lenguaje de programación Java usando un modelo MVC para su arquitectura, en cuanto a la implementación de la interfaz se ha realizado mediante XML. El segundo módulo es una API del lado del servidor que permite la conexión de la aplicación con la base de datos, se ha implementado usando la tecnología PHP. El tercer y último módulo consiste en la base de datos, se ha utilizado la tecnología MySQL para su creación y gestión, ya que es el lenguaje apropiado para una base de datos relacional.

La aplicación superó las pruebas, la validación y la evaluación finales por parte del usuario final, aunque se han detectado mejoras en las que se podrá trabajar para el futuro.

Abstract

There are currently lots of families with pets at home. Their increase has gone along with the increasing concern of those families, and in general of society, for their pets' health. It is, thus, necessary to carefully schedule their veterinary appointments, their vaccines, etc. what sometimes implies to collect and handle lots of documents.

My Bachelor's Thesis is motivated by this situation. After studying the state of the art, we have defined the goals of a native application for Android (Ourpets) to manage our pets' relevant medical information

This application has been developed by means of three modules for three different functionalities. The first one includes the mobile application itself and the user interface, that have been respectively implemented with Java (and the help of a MVC model for the system architecture) and XML. The second one is a server API to connect the application and the data base. This module has been implemented with PHP technology. Finally, the third module is the data base, managed with MySQL technology.

Ourpets has been successfully tested, and it passed the validation and final user's evaluation; although some improvements have been detected that could be tackled in the future.

Palabras clave

Ourpets, servidor, aplicación nativa, aplicación móvil, Android, base de datos relacional, MySQL, PHP, Java, AWS, XML, Android Studio, MySQL Workbench.

Keywords

Ourpets, server, native application, mobile application, Android, relational database, MySQL, PHP, Java, AWS, XML, Android Studio, MySQL Workbench

Agradecimientos

Quiero agradecer en primer lugar a mi tutor Alfonso Ortega De La Puente, la confianza puesta en mí al aceptar mi propuesta de TFG, así como toda la ayuda prestada durante el desarrollo del mismo.

Agradecer también a mis compañeros y amigos por hacerme sonreír siempre incluso en los momentos más difíciles. A Patricia por aguantarme tantos años y apoyarme en todo momento.

Agradecer a todos mis compañeros de mi equipo de rugby y en especial a mis entrenadores por enseñarme valores muy importantes y útiles en el día a día de mi vida.

Y lo más importante agradecer a mis padres todo lo que han hecho por mí y por enseñarme a no rendirme nunca y perseguir mis sueños por difíciles que sean.

INDICE DE CONTENIDOS

1 Introducción.....	1
1.1 Marco del proyecto	1
1.2 Motivación del proyecto.....	1
1.3 Objetivo	1
1.4 Estructura del documento	2
2 Estado del arte.....	3
2.1 Introducción.....	3
2.2 Sistemas existentes	4
2.2.1 WAP Diary – Beta	4
2.2.2 Ficha animal VET	5
2.2.3 Mi veterinario	6
2.2.4 11 Pets	7
2.2.5 Dog Health	8
2.3 Conclusiones.....	9
3 Objetivos/Funcionalidades.....	11
3.1 Objetivos Genéricos	11
3.2 Objetivos Específicos y Funcionales	11
3.2.1 Alta de mascotas	11
3.2.2 Alta y recordatorio de vacunas	11
3.2.3 Alta y recordatorio de desparasitaciones	12
3.2.4 Registro y recordatorio de visitas al veterinario.....	12
3.2.5 Control de peso	12
3.2.6 Veterinarios más cercanos	12
3.3 Proceso Funcional.....	12
4 Definición del proyecto.....	15
4.1 Metodología.....	15
4.2 Herramientas usadas	16
4.2.1 Android Studio.....	17
4.2.2 MySQL	17
4.2.3 XML	17
4.2.4 Java.....	17
4.2.5 JSON.....	18
4.2.6 Librería Volley	18
4.2.7 Librería MPAndroidChart.....	18
4.2.8 Amazon Web Services (AWS).....	18
5 Análisis y diseño	19
5.1 Análisis	19
5.1.1 Requisitos Funcionales	19
5.1.2 Requisitos No Funcionales	20
5.2 Diseño	20
5.2.1 Base de datos	21
5.2.2 Aplicación móvil.....	22
5.2.3 Servidor	23
6 Implementación	25
6.1 Estructura de la aplicación	25
6.2 Cliente (Aplicación)	25
6.3 Servidor.....	27

6.4 Base de datos	28
6.5 Interfaz de usuario.....	28
6.5.1 Registro del usuario	29
6.5.2 Login del usuario	30
6.5.3 Listado y alta de mascotas	31
6.5.4 Detalles de la mascota	32
6.5.5 Vacunas	33
6.5.6 Alta e histórico de desparasitaciones	34
6.5.7 Alta e histórico de consultas al veterinario	35
6.5.8 Alta y gráfica de pesos	36
6.5.9 Veterinarios más cercanos	37
6.5.10 Ayuda	38
6.5.11 Navegación	38
7 Pruebas	39
7.1 Estrategia de pruebas	39
7.1.1 Pruebas unitarias	39
7.1.2 Pruebas de integración	39
7.1.3 Pruebas de interfaz	40
7.2 Resultados de las pruebas	40
7.2.1 Pruebas unitarias	40
7.2.2 Pruebas de integración	40
7.2.3 Pruebas de interfaz	40
8 Conclusiones y trabajo futuro.....	43
8.1 Conclusiones.....	43
8.2 Trabajo futuro	43
Referencias	45
Glosario.....	47

INDICE DE FIGURAS

Figura 4-1: Desarrollo en cascada.....	15
Figura 5-1: Arquitectura de Ourpets.....	21
Figura 4-1: Flujo de operación de Ourpets.....	23
Figura 6-1: Estructura de cliente.....	27
Figura 6-2: Diagrama implementación de la base de datos.....	28
Figura 6-3: Registro en Ourpets.....	29
Figura 6-4: Login en Ourpets.....	30
Figura 6-5: Listado de mascotas.....	31
Figura 6-6: Alta de mascotas.....	31
Figura 6-7: Destalles de la mascota.....	32
Figura 6-8: Vacunas disponibles.....	33
Figura 6-9: Histórico de vacunas.....	33
Figura 6-10: Listado de desparasitaciones.....	34
Figura 6-11: Alta de consultas en veterinario.....	35
Figura 6-12: Historico de consultas en veterinario.....	35
Figura 6-13: Alta de pesos.....	36
Figura 6-14: Gráfico de pesos.....	36
Figura 6-15: Veterinarios mas cercanos.....	37
Figura 6-16: Navegación en Ourpets.....	38

1 Introducción

1.1 Marco del proyecto

Este proyecto ha sido realizado por el estudiante Iván Zorita Bote, como propuesta personal. El proyecto consta de una aplicación móvil, nativa de Android, que pretende facilitar la gestión del historial veterinario para los usuarios con mascotas.

1.2 Motivación del proyecto

Hoy en día es muy común ver familias con mascotas en casa, y con el paso del tiempo la salud de los animales se está cuidando con más detalle. Esto implica vacunaciones, desparasitaciones y visitas al veterinario periódicas.

El objetivo de esta aplicación es facilitar que el propietario de una mascota pueda acceder a toda la información sobre la salud de su mascota siempre que lo necesite sin necesidad de tener que buscar en los papeles, de forma gratuita. Proporcionando además la posibilidad de programar recordatorios sobre revisiones y tomas de medicamentos.

1.3 Objetivo

El objetivo de este proyecto es la definición e implementación de una aplicación nativa de Android que facilite la gestión de la información sobre las mascotas de cualquier usuario.

El resultado es una aplicación con la que el usuario podrá almacenar información sobre todas mascotas que desee.

La aplicación también servirá para programar recordatorios de revisiones en el veterinario, desparasitaciones y próximas vacunas. Así como programar una alarma periódica para recordar las tomas de un medicamento que se le haya mandado a la mascota.

La aplicación se ha realizado haciendo uso del IDE Android Studio, PHP para implementar los servicios del lado del servidor y MySQL para la gestión de una base de datos relacional.

1.4 Estructura del documento

La memoria consta de las siguientes secciones:

- Sección 2: se hace un análisis del estado del arte, es decir, un análisis de sistemas similares a Ourpets que ya existan en el mercado.
- Sección 3: se hace una descripción de los objetivos genéricos y funcionales que debe cumplir la aplicación en su implementación.
- Sección 4: se hace una definición el proyecto, así como las tecnologías de las que se han hecho uso durante el desarrollo.
- Sección 5: se hace un análisis de los requisitos funcionales y no funcionales que debe cumplir la aplicación.
- Sección 6: se hace una descripción primero la estructura tanto de la aplicación como del servidor. Después de eso se describirá la implementación de la aplicación y se explicará con detalle la interfaz de usuario.
- Sección 7: se hace una descripción la implementación del plan de pruebas y los resultados obtenidos durante ellas.
- Sección 8: se hace una explicación las conclusiones obtenidas durante el desarrollo y las conclusiones finales. Además, también se explicarán las líneas de trabajo futuras, para continuar el trabajo en la mejora de la aplicación.
- Sección 9: contiene las referencias utilizadas en el desarrollo de la memoria.

2 Estado del arte

2.1 Introducción

Con el paso de los años el número de familias con mascotas y la cantidad de mascotas en cada casa se ha ido incrementado. Tradicionalmente la información sobre la mascota se ha guardado en papel o en el peor de los casos no se guardaba. Esto supone tener que reservar espacio para los papeles, que se pierdan algunos de ellos, posibilidades de que se olvide una cita con el veterinario, etc.

El principal problema de guardar todos los papeles en carpetas, es su ineficiencia. Ante la necesidad de buscar algún dato sobre una mascota requiere remover un montón de papeles, o la imposibilidad de encontrarlo si te encuentras fuera de casa, como por ejemplo en el veterinario.

Con la aparición de los Smartphone y las aplicaciones móviles, ha permitido el desarrollo de aplicaciones que permitan guardar parte de esta información. Con ello cualquier persona desde cualquier lugar podría consultar rápidamente un dato sobre su mascota.

Como solución a este problema, este Trabajo de Fin de Grado tiene como objetivo la creación de Ourpets, una aplicación móvil Android, que tiene como objetivo recopilar toda la información relevante sobre las mascotas de la manera mas eficiente, además de poder programar notificaciones que avisen de vacunas, desparasitaciones o revisiones próximas. Para ello se ha realizará un análisis de un conjunto de aplicaciones cuya funcionalidad sea similar a la definida en el proyecto.

2.2 Sistemas existentes

A continuación, se va a proceder al estudio algunas de las aplicaciones ya existentes.

2.2.1 WAP Diary – Beta

Aplicación disponible en el Play Store de forma gratuita para dispositivos Android [1].

Esta diseñada por veterinarios lo que le permite tener un mayor conocimiento sobre la información importante para la salud de la mascota.

Únicamente deja agregar un animal gratuitamente, por cada animal extra que se quiera añadir se debe pagar o invitar a un amigo en Facebook para que descargue la aplicación y que éste confirme la invitación.



La aplicación contiene poca información acerca de posibles vacunas que se le pueden poner a tu mascota, sólo deja guardar información sobre las principales de cada mascota.

Las desparasitaciones que puedes almacenar también son limitadas, sólo se puede guardar información sobre una desparasitación interna y otra externa.

No guarda ningún tipo de información acerca de visitas al veterinario ni un control del peso de la mascota.

2.2.2 Ficha animal VET

La descarga de la aplicación es gratuita y está disponible en el Play Store para dispositivo Android [2].

A pesar de que la descarga es gratuita, para desbloquear todas las funcionalidades de la aplicación es necesario pagar.



En la aplicación no hace falta registrar una cuenta, personal. Primero se deben dar de alta propietarios, y después se dará de alta a las mascotas.

La versión gratuita de la aplicación únicamente deja guardar información sobre las consultas al veterinario.

No permite programar notificaciones ni alarmas sobre revisiones o citas futuras.

Una vez desbloqueada toda la funcionalidad permite guardar información sobre vacunas, fármacos, exámenes y cirugías.

La aplicación almacena los datos en local haciendo uso de SQLite, con lo cual si se hace una limpieza o se cambia de móvil se pierden todos los datos guardados sobre las mascotas.

En ninguna versión de la aplicación deja guardar información sobre las desparasitaciones, ni programar alertas sobre ellas.

2.2.3 Mi veterinario

Está disponible tanto en plataforma web, como en dispositivos móviles tanto iOS como Android. Es gratuita en cualquiera de sus plataformas. Aunque es exclusiva de clientes de veterinarios Winvet [3].



La aplicación no guarda información ni de desparasitaciones, ni vacunas, ni control de peso de la mascota.

Se puede guardar información básica de la mascota, como si esta esterilizado.

La aplicación permite solicitar citas a uno de los veterinarios Winvet, y programar recordatorios sobre las mismas.

Los veterinarios pueden publicar noticias y consejos, que le llegarán al propietario a través de la aplicación.

Se pueden guardar todas las mascotas que requiera el usuario sin costes adicionales.

La aplicación está disponible en varios idiomas como español, inglés, catalán, etc.

2.2.4 11 Pets

Está disponible de forma gratuita en el Play Store para móviles Android [4].

Permite guardar recordatorios para cuidados básicos de la mascota, como por ejemplo lavado de dientes, baño, corte uñas y corte de pelo.



Permite crear una galería de imágenes de la mascota.

Se puede crear un historial clínico completo de la mascota incluyendo todas las pruebas clínicas realizadas a la mascota, patologías existentes, alergias, test genéticos y cirugías.

Se pueden programar alertas periódicas para las desparasitaciones y las vacunas.

Podremos guardar los medicamentos que debe tomar el animal guardando las dosis y el periodo de tiempo con el que se la debe tomar.

Permite llevar el control del peso de la mascota, así como la temperatura y la altura.

Tiene una pantalla con un calendario donde se puede revisar las alertas programadas y editarlas rápidamente.

En un principio no necesita de registro, por los datos se guardarán en local, es decir si se borran los datos de la aplicación, se perderán los datos de la mascota.

Se ofrece una opción para realizar un back up de los datos y restaurarlos. Para poder hacer uso de ella es necesario registrarse en la aplicación o en la página web.

2.2.5 Dog Health

Aplicación disponible sólo en Play Store para dispositivos Android [5].

Permite guardar información sobre las desparasitaciones, así como programar notificaciones periódicas con la primera y la última toma.



Guarda información sobre las vacunaciones y rellenando el campo fecha recordatorio añade una notificación en la fecha indicada.

El sistema de almacenamiento de visitas y cirugías guarda el mismo formato que las vacunas, se rellena la información y ofrece la posibilidad de añadir una notificación en una fecha determinada.

Podremos realizar una búsqueda sencilla de los veterinarios más cercanos y acceder a Google Maps para poder ver los datos de contacto.

En un inicio no es necesario el registro y guarda los datos en SQLite de forma local en la memoria del teléfono. Si se borran los datos se pierde toda la información almacenada de las mascotas.

Se puede hacer un back up si se tiene una cuenta de Google Drive, permitiendo también restaurar los datos.

2.3 Conclusiones

Después de realizar el estudio del estado del arte, se han podido detectar algunas necesidades no cubiertas y posibles mejoras en algunas funcionalidades ya implementadas en las aplicaciones.

En la mayoría de las aplicaciones no se puede almacenar el control del peso de la mascota, necesario para mantener la mascota sana y prevenir problemas físicos.

En casi todas las aplicaciones es necesario algún método de pago restringiendo así el uso de la funcionalidad completa de la aplicación.

En la mayoría de las aplicaciones se carece de un método de búsqueda rápida de los veterinarios más próximos. Esto es importante ya que en caso de emergencia se debe poder localizar un veterinario cercano y ver cómo llegar.

En un único caso se permite concertar citas con los veterinarios directamente por la aplicación. Pero esta queda restringida al uso exclusivo de los veterinarios Winvet.

En la mayoría de los casos el uso de notificaciones y recordatorios o no existe o es muy ineficiente. Siendo una funcionalidad de muy útil para el usuario, permitiéndole despreocuparse hasta que la fecha de una revisión o vacuna se acerca.

3 Objetivos/Funcionalidades

En este apartado vamos a explicar los distintos objetivos y funcionalidades satisfechos por la aplicación Ourpets.

3.1 Objetivos Genéricos

El objetivo de este proyecto es el diseño y desarrollo de un sistema de gestión de mascotas. Este sistema consistirá en una aplicación móvil en Android.

3.2 Objetivos Específicos y Funcionales

En esta sección se va a describir los objetivos específicos y los subsistemas de la aplicación. La división en subsistemas se realiza satisfaciendo las distintas funcionalidades de la aplicación, es decir, cada subsistema satisface una funcionalidad distinta de Ourpets.

3.2.1 Alta de mascotas

Esta subsección satisface la funcionalidad más básica de la aplicación, que es el alta de mascotas. Para dar de alta una mascota se debe rellenar un formulario con los datos de la mascota, incluido el chip para perros y gatos. Opcionalmente se puede subir una foto de la mascota.

Una vez dada de alta una mascota se desbloquean el resto de funcionalidades de la aplicación.

3.2.2 Alta y recordatorio de vacunas

Ourpets ofrece la posibilidad de llevar un control histórico de vacunas. Primero se selecciona la vacuna que se le ha puesto a la mascota, se redirige a un formulario donde se rellena la fecha y opcionalmente se puede rellenar un campo con la fecha de la próxima vacuna. La aplicación automáticamente genera una notificación que se mostrara el día de la próxima vacunación.

3.2.3 Alta y recordatorio de desparasitaciones

Se podrá llevar un control histórico de las desparasitaciones. Para dar de alta una desparasitación se debe rellenar un formulario con los datos de las desparasitaciones. Se ofrece la posibilidad de rellenar una fecha de próxima desparasitación, lo que genera automáticamente una notificación el día de la próxima desparasitación.

3.2.4 Registro y recordatorio de visitas al veterinario

Ourpets ofrece un registro histórico de las visitas al veterinario con su diagnóstico. Para registrar una visita se rellenará un formulario con los detalles de la visita, incluido el precio. Opcionalmente puede rellenar el campo *próxima revisión*, que genera una notificación el día de la próxima visita. También se ofrece la posibilidad de programar recordatorios periódicos para dar la medicación, en caso de que sea necesaria, a la mascota.

3.2.5 Control de peso

Ourpets ofrece la posibilidad de llevar el control del peso de la mascota. El usuario puede introducir el peso de la mascota. En una tabla se mostrarán los últimos 10 peso introducidos. Cuando se rellena el primer dato con el peso se comienza a generar una gráfica con el histórico de pesos de la mascota.

3.2.6 Veterinarios más cercanos

Se podrán consultar los veterinarios más cercanos a tu posición actual. En esta sección se ofrece un listado con los veterinarios más cercanos además de poder utilizar Google Maps para ver la ruta hasta el seleccionado.

3.3 Proceso Funcional

El usuario debe registrarse en la aplicación para poder usarla. Una vez creada la cuenta podrá acceder y crear mascotas.

Una vez el usuario ha accedido se muestra el listado de las mascotas guardadas, si las hubiese. Se puede añadir una mascota rellenando un formulario con los datos de la misma.

Cuando el usuario entra en los detalles de la mascota ya puede navegar por las distintas funcionalidades de la aplicación. Para ello hará uso del sistema de navegación que se encuentra en la esquina superior izquierda.

El usuario podrá dar de alta y programar recordatorios para vacunas, desparasitaciones y visitas al veterinario. Así como consultar el histórico de pesos y los veterinarios más cercanos a nuestra posición.

4 Definición del proyecto

4.1 Metodología

Para el desarrollo de la aplicación se ha usado un método de desarrollo en cascada iterativo, se muestra en la imagen 4.1. Se ha utilizado este tipo de desarrollo por la necesidad de integrar gran número de funcionalidad.

Este tipo de desarrollo permite verificar al final de cada fase si el proyecto está listo para avanzar a la siguiente, de esta forma se pueden detectar errores antes de llegar a la fase de pruebas y puede ser subsanado previamente.

Este tipo de proceso permite poder incorporar desarrollo de nuevas funcionalidades e integrarlas de forma más eficiente que otros tipos de desarrollo.

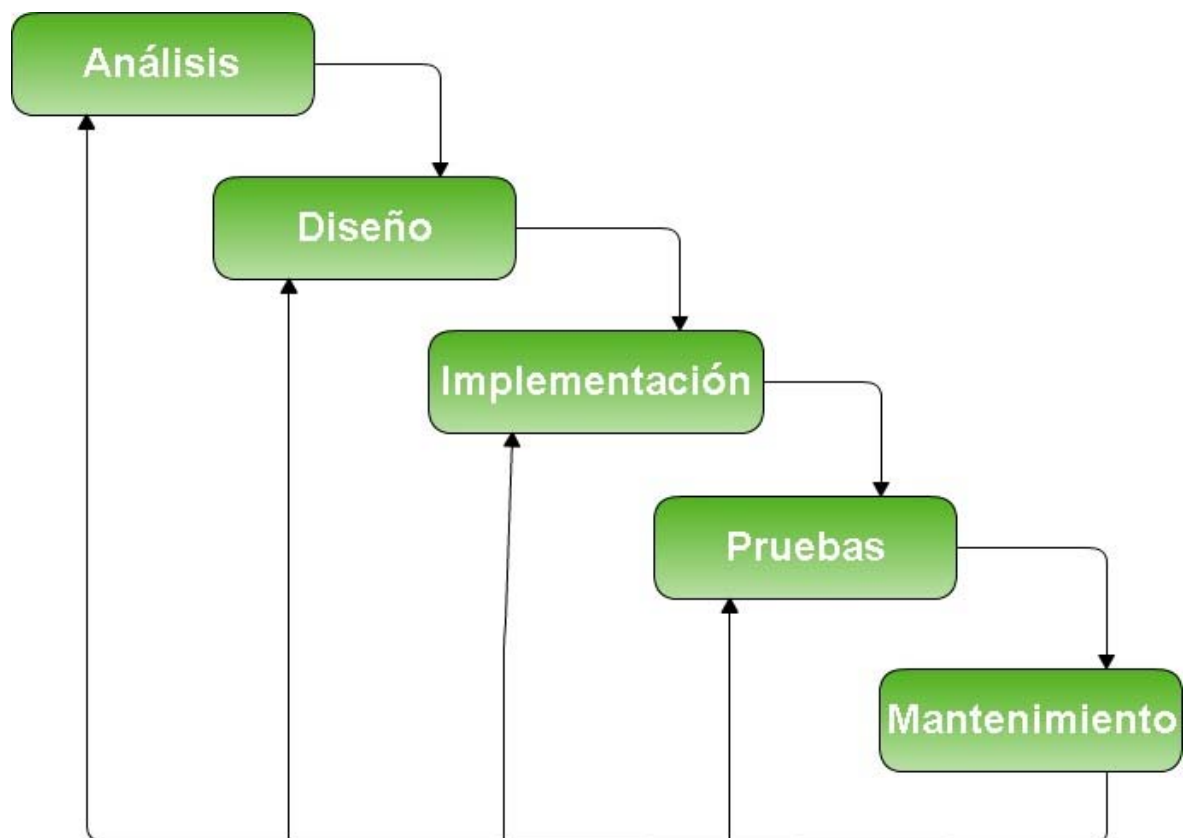


Figura 4-1: Desarrollo en cascada

A continuación, se van a enumerar las actividades realizadas divididas en las fases del ciclo de vida en cascada.

Análisis

- Estudio del estado del arte.
- Especificación de objetivos y funcionalidades.
- Especificación de requisitos funcionales y no funcionales.

Diseño

- Diseño de la base de datos
- Diseño de la arquitectura de la aplicación
- Diseño de los módulos de la aplicación
- Diseño de las interfaces de usuario

Implementación

- Implementación de la base de datos MySQL
- Implementación de la aplicación en Android
- Implementación del servidor en PHP

Pruebas y validación

- Desarrollo del plan de pruebas
- Diseño de casos de prueba
- Realización de casos de prueba
- Evaluación de los resultados obtenidos en un caso de prueba real

4.2 Herramientas usadas

Para el desarrollo de Ourpets se ha usado las siguientes herramientas.

- IDE de desarrollo de aplicaciones móviles de Android
- Base de datos relacional MySQL
- Gestor de bases de datos relacionales MySQL Workbench
- Lenguaje de marcas: XML
- Lenguaje de programación: Java
- Notación de objetos en Javascript: JSON
- Librería de Android para realizar peticiones Http: Volley
- Librería de Android para generar gráficos: MPAndroidChart
- Contenedor de servidores web: Amazon Web Services

Se va a explicar brevemente las herramientas utilizadas.

4.2.1 Android Studio

Android Studio es un IDE que permite el desarrollo de aplicaciones móviles nativas en Android [6].

Es necesario para el desarrollo de la aplicación Ourpets, ya que se trata de una aplicación nativa en Android. Android Studio facilita todo el desarrollo de la aplicación proporcionando librerías y API de Android nativas.

4.2.2 MySQL

El modelo de datos que se va a usar es relacional, por lo que vamos a hacer uso de MySQL [7].

MySQL es un sistema de bases de datos relacional que usa como lenguaje de acceso SQL (Structure Query Language).

Para poder gestionar la base de datos en local, se hizo uso de XAMPP y PhpMyAdmin. Al migrar la base de datos a un servidor web cambiamos el programa de gestión de acceso a la base de datos a MySQL Workbench [8].

MySQL Workbench es una herramienta visual de diseño de bases de datos que permite la diseño, creación, administración y mantenimiento de bases de datos MySQL.

4.2.3 XML

XML es un lenguaje de marcas. La creación de aplicaciones nativas en Android Studio hace necesario XML. XML es el lenguaje principal utilizado por Android para crear las interfaces de usuario, los archivos de recursos, arrays, etc.

Ourpets ha sido desarrollada como una aplicación nativa de Android por lo que es necesaria la utilización de XML.

4.2.4 Java

Java es un lenguaje de programación orientado a objetos. Android usa como lenguaje nativo Java unido a las diferentes funciones proporcionadas por la API de Android.

Ourpets ha sido desarrollada como una aplicación nativa de Android por lo que es necesaria la utilización de Java [9].

4.2.5 JSON

JSON, acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos [10]. De ha usado JSON ya que puede ser interpretado por cualquier lenguaje de programación. Se esta forma si cambiásemos el lenguaje de programación del servidor, en la parte de la aplicación no haría falta cambiar nada.

4.2.6 Librería Volley

Volley es una librería desarrollada por google que permite realizar peticiones Http a una aplicación Android.

Ourpets necesita realizar peticiones Http al servidor externo donde está alojada la base de datos, para ello hemos elegido volley debido a su simplicidad de uso y al control de errores proporcionado por la librería.

4.2.7 Librería MPAndroidChart

La librería MPAndroidChart permite generar gráficos de forma sencilla dentro de una aplicación Android.

Ourpets genera una graficar para controlar los pesos, para ello se ha utilizado MPAndroidChart, debido a su gran variedad de gráficos y a las animaciones que se pueden utilizar.

4.2.8 Amazon Web Services (AWS)

Durante el desarrollo de la aplicación se hizo uso de XAMPP como servidor local. Una vez terminada la aplicación se migró la base de datos y el código de servidor a un gestor web: AWS [11].

AWS es una plataforma de servicios en la nube que ofrece almacenamiento de bases de datos, entrega de contenido y otras funcionalidades.

5 Análisis y diseño

5.1 Análisis

Después de realizar el análisis del estado del arte, y realizar un análisis de las necesidades del usuario, se realizará el análisis de los requisitos funcionales y no funcionales de la aplicación.

5.1.1 Requisitos Funcionales

RF.1 Ourpets permite al usuario registrarse almacenando una serie de datos personales.

RF.2 Ourpets requiere del ingreso de un usuario y contraseña para poder logarse en la aplicación.

RF.3 El usuario podrá almacenar tantas mascotas como desee, introduciendo una serie de detalles acerca de la misma.

RF.4 El usuario no podrá hacer uso de toda la funcionalidad de la aplicación hasta que no ingrese su primera mascota.

RF.5 El usuario podrá modificar y eliminar la mascota.

RF.6 El usuario podrá añadir, editar y eliminar vacunas, y recordatorios de futuras vacunas.

RF.7 El usuario podrá añadir, editar y eliminar desparasitaciones, y recordatorios de revisiones y futuras desparasitaciones.

RF.8 El usuario podrá añadir, editar y eliminar visitas veterinarias, y recordatorios para las revisiones.

RF.9 El usuario podrá añadir recordatorios para las tomas de los medicamentos de la mascota.

RF.10 El usuario podrá llevar el control del peso del animal añadiendo el peso actual y la fecha. Se le mostrará una gráfica con la evolución del peso.

RF.11 El usuario podrá localizar los veterinarios más cercanos. Se le mostrarán por orden de más cercano a más lejano.

5.1.2 Requisitos No Funcionales

RNF.1 El idioma de la aplicación será el castellano

RNF.2 La aplicación se podrá usar desde cualquier dispositivo Android, quedando excluidos los dispositivos con sistema operativo distinto.

RNF.3 El usuario debe identificarse mediante usuario y contraseña.

RNF.4. En las distintas notificaciones se informará del motivo que ha generado dicha notificación.

RNF.5 En caso de que la aplicación genere algún tipo de error será notificado mediante un mensaje.

RNF.6 Es necesaria la conexión a internet durante el uso de la aplicación, si la conexión fuese mala e impidiese el uso de la aplicación se notificará al usuario con un mensaje.

5.2 Diseño

Una vez realizado el análisis funcional, se puede proceder al diseño de la aplicación. El diseño se puede dividir en tres módulos distintos: la aplicación, la base de datos y la parte del servidor. La base de datos y el servidor se encuentran alojados en AWS por lo que lo podemos agrupar dentro de un mismo modulo, aunque las implementaciones sean distintas. Podemos observar la distribución de la aplicación en la figura 5.1.

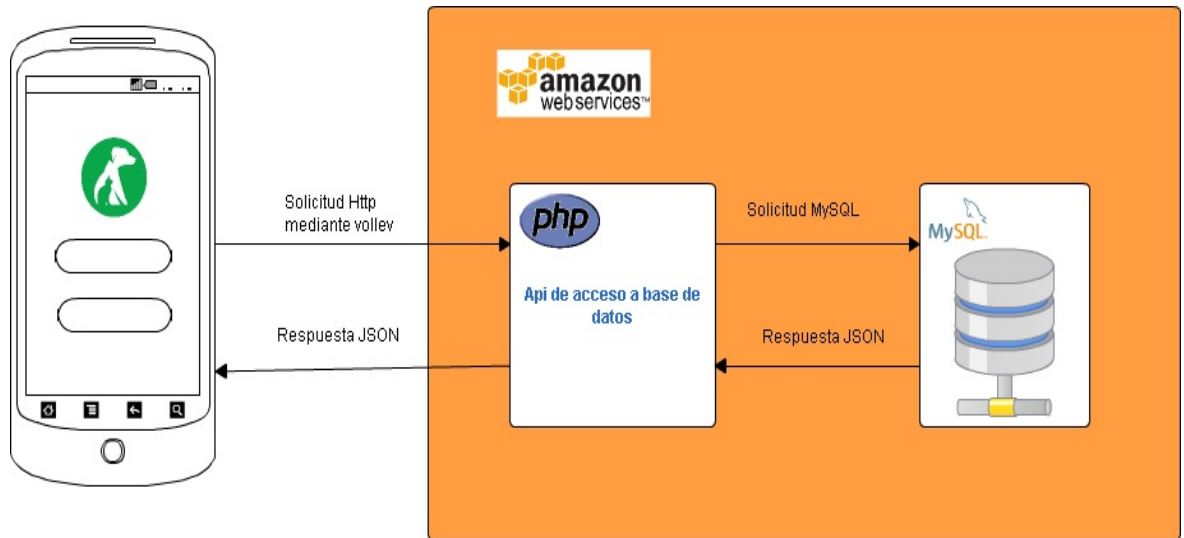


Figura 5-1: Arquitectura Ourpets 1

Los principales módulos que podemos observar son los siguientes:

- Base de datos relacional MySQL en la que se guarda toda la información sobre los usuarios, mascotas, vacunas, etc.
- Aplicación móvil. Este módulo consta de la lógica de la aplicación y la interfaz de usuario. La aplicación móvil se encarga de comunicarse con el servidor mediante llamadas Http realizadas con la librería Volley. La respuesta se puede recibir en distintos lenguajes, ya que Volley admite múltiples formatos de respuesta, pero el servidor nos envía la información en formato JSON para su procesamiento en la aplicación.
- Servidor PHP. Recibe una petición Http con parámetros desde la aplicación móvil. Procesa la petición y realiza una petición a la base de datos MySQL solicitando la información requerida por la aplicación.

El servidor también se encarga de codificar la respuesta en formato JSON y se la envía a la aplicación para que realice las operaciones pertinentes con ella.

5.2.1 Base de datos

La base de datos utilizada por Ourpets es una base de datos relacional implementada en MySQL.

Las tablas de la base de datos son las siguientes:

- Usuarios: almacena la información sobre los usuarios que se han registrado.
- Mascotas: almacena toda la información de una mascota, así como el identificador del usuario que la ha registrado.
- Vacunas: contiene una lista de vacunas disponible para seleccionar cuando vayamos a registrar la vacunación de la mascota.
- Vacuna_mascota: guarda todos los registros de las vacunaciones de las mascotas.
- Desparasitaciones: guarda todos los registros sobre las desparasitaciones de las mascotas.
- Peso: guarda todos los detalles de las mediciones de los pesos realizadas a las mascotas.
- Visitas: guarda toda la información de las visitas al veterinario realizadas por el usuario con sus mascotas.

5.2.2 Aplicación móvil

La aplicación se divide en dos partes, la interfaz de usuario y la lógica de la aplicación.

El diseño de la interfaz se ha guiado por su simplicidad y limpieza, haciendo que el uso de la aplicación resulte sencillo para el usuario.

En la figura 5-2 podemos apreciar el flujo de operación seguido por la aplicación móvil.

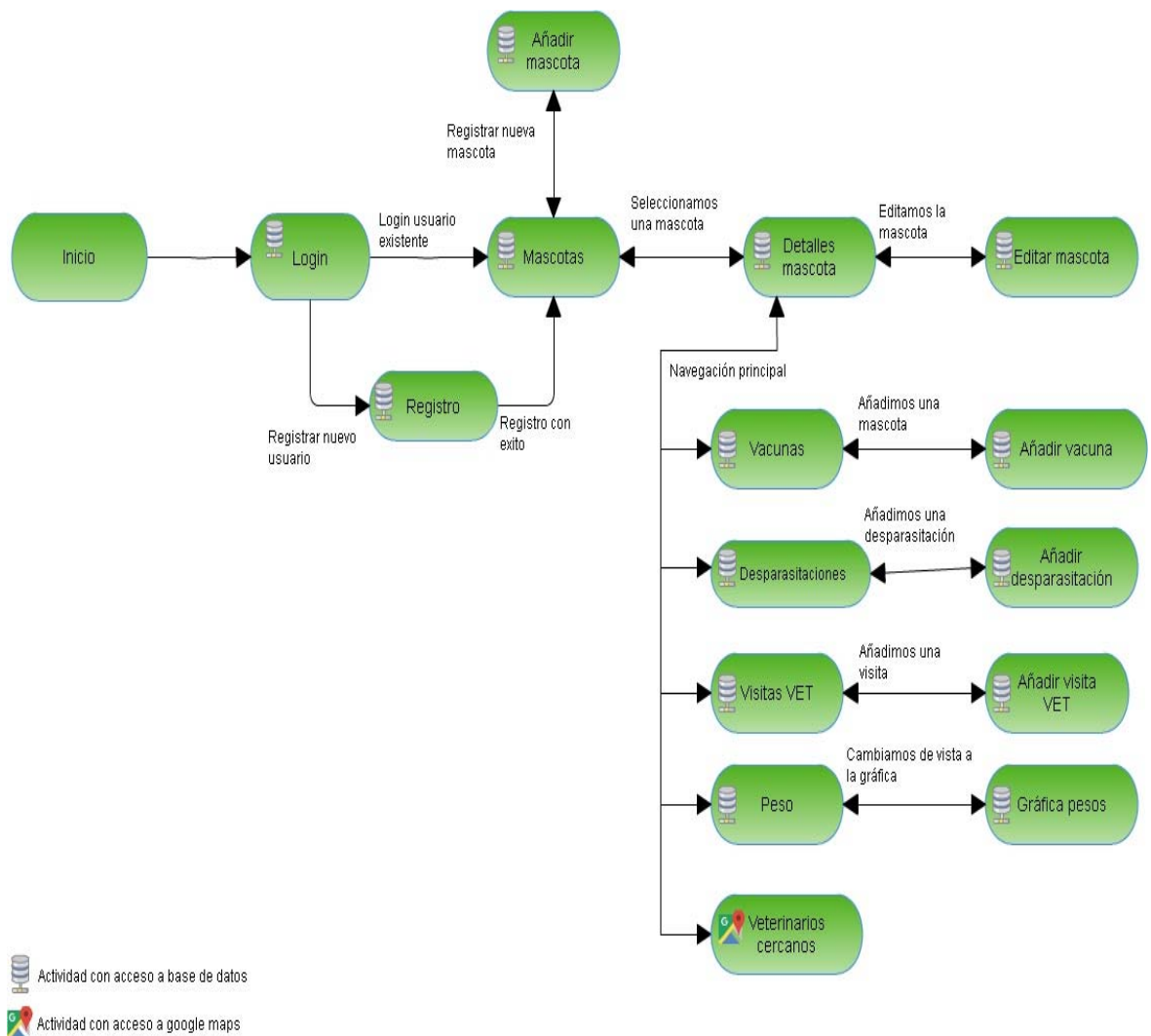


Figura 5-2: Flujo de operación Ourpets

5.2.3 Servidor

La implementación del servidor es independiente al cliente (Aplicación móvil), de tal forma que se pueden conectar varios usuarios simultáneamente. Este servidor contiene los servicios necesarios para que se comuniquen el cliente y la base de datos, ya que en una aplicación Android no es posible conectar directamente la aplicación con la base de datos.

6 Implementación

En esta sección se va a explicar cómo se ha implementado la aplicación, así como las tecnologías utilizadas en ella.

6.1 Estructura de la aplicación

Como se ha visto en la sección de diseño la aplicación se divide en tres módulos: la parte de la aplicación (cliente), la parte del servidor y la base de datos.

La aplicación ha sido desarrollada en Android usando el IDE de desarrollo de Android Studio. La aplicación se estructura según el modelo MVC (Modelo Vista Controlador). La parte de la vista se ha codificado usando XML y una clase Java para retocar las imágenes. La parte del controlador, que es la lógica de la aplicación, se ha codificado usando Java y funciones proporcionadas por la API de Android.

El servidor alojado en AWS (Amazon Web Services) es el encargado de comunicar la aplicación con la base de datos, ya que actualmente no se puede conectar la aplicación directamente con una base de datos externa. Se ha codificado haciendo uso de PHP, y se distribuye usando un fichero PHP para cada servicio requerido por la aplicación.

La base de datos sigue una estructura relacional entre sus tablas, por lo que hemos elegido MySQL para su creación y usaremos las funciones de MySQL para comunicar el servidor con la base de datos.

6.2 Cliente (Aplicación)

La estructura de la aplicación sigue el modelo MVC, aunque la mayoría de la lógica de la aplicación se encuentra en la parte del controlador. También se encuentra parte de la lógica en el modelo, donde se guardan y manejan los datos que vienen de la base de datos.

La aplicación hace uso de una actividad para cada pantalla de la interfaz de usuario, aunque se ha programado una clase abstracta que implementa toda la funcionalidad conjunta de las distintas actividades y declara las funciones implementadas en el resto de actividades.

Cada actividad contiene como mínimo un fragmento, un fragmento sigue la misma lógica de una actividad, pero ofrece la posibilidad de, en una misma pantalla, poder tener dos o más fragmentos. Esto puede ser muy útil a la hora de adaptar la aplicación a tabletas, que con una pantalla más grande puede dividirla en dos y cargar dos fragmentos con dos funcionalidades distintas en la misma pantalla.

Para realizar las llamadas al servidor se ha implementado una clase estática, que se puede llamar desde cualquier actividad cuando se requieran datos guardados en la base de datos.

La aplicación sigue la estructura típica de una aplicación nativa de Android que sigue un modelo MVC.

Dentro del directorio java se encuentra el código Java que se encargará de toda la lógica implementada en la aplicación. Como se puede ver en la figura 6-1 se han creado tres directorios llamados modelo, vista y controlador. Cada uno de ellos contiene las clases propias de cada componente del modelo MVC.

Dentro del directorio res se encuentran todos los recursos de la aplicación. Los directorios más importantes de res son:

- Layout: en este directorio se guardan todos los XML que pintan las pantallas de la interfaz de usuario.
- Anim: este directorio guarda los XML con todas las animaciones de la aplicación.
- Drawable: este directorio agrupa todas las imágenes que se mostrarán en la aplicación. Normalmente las imágenes se guardan en 5 resoluciones distintas: mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi. Ordenadas de menor a mayor resolución.
- Menu: guarda todos los ficheros XML de todos los menús de la aplicación.
- Mipmap: en mipmap se guardan las imágenes para el icono de la aplicación.
- Values: en values guardan los XML con todos los recursos de la aplicación, por ejemplo, las cadenas de caracteres, los colores, etc.

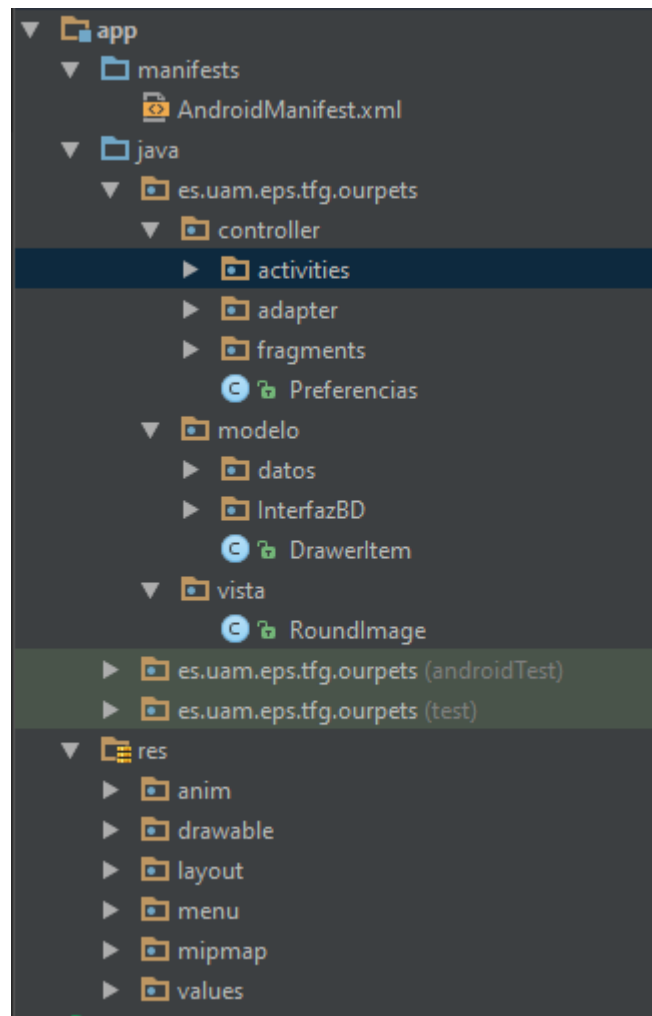


Figura 6-1: Estructura del cliente.

6.3 Servidor

El servidor sirve de pasarela entre la aplicación y la base de datos, ya que las aplicaciones móviles nativas en Android no se pueden comunicar directamente con la base de datos.

La capa del servidor se ha implementado en PHP que es un lenguaje de programación que trabaja del lado del servidor. La estructura es muy simple ya que consta de un directorio que contiene los ficheros php y un subdirectorio para almacenar las fotos que suben los clientes.

Se ha implementado un fichero PHP para cada servicio que solicita el cliente (la aplicación móvil). El cliente solicita, mediante una petición http, unos datos de las bases de datos. El servidor crea una consulta MySQL y realiza la petición a la base de datos. Una vez se ha obtenido la respuesta de la base de datos, se codifica usando JSON y se manda a la aplicación para que la interprete y realice las operaciones pertinentes con los datos de respuesta del servidor.

6.4 Base de datos

En la figura 6-2 podemos ver la estructura de la base de datos. Desde el servidor se accede a la base de datos haciendo uso de las funciones propias MySQL que ofrece PHP.

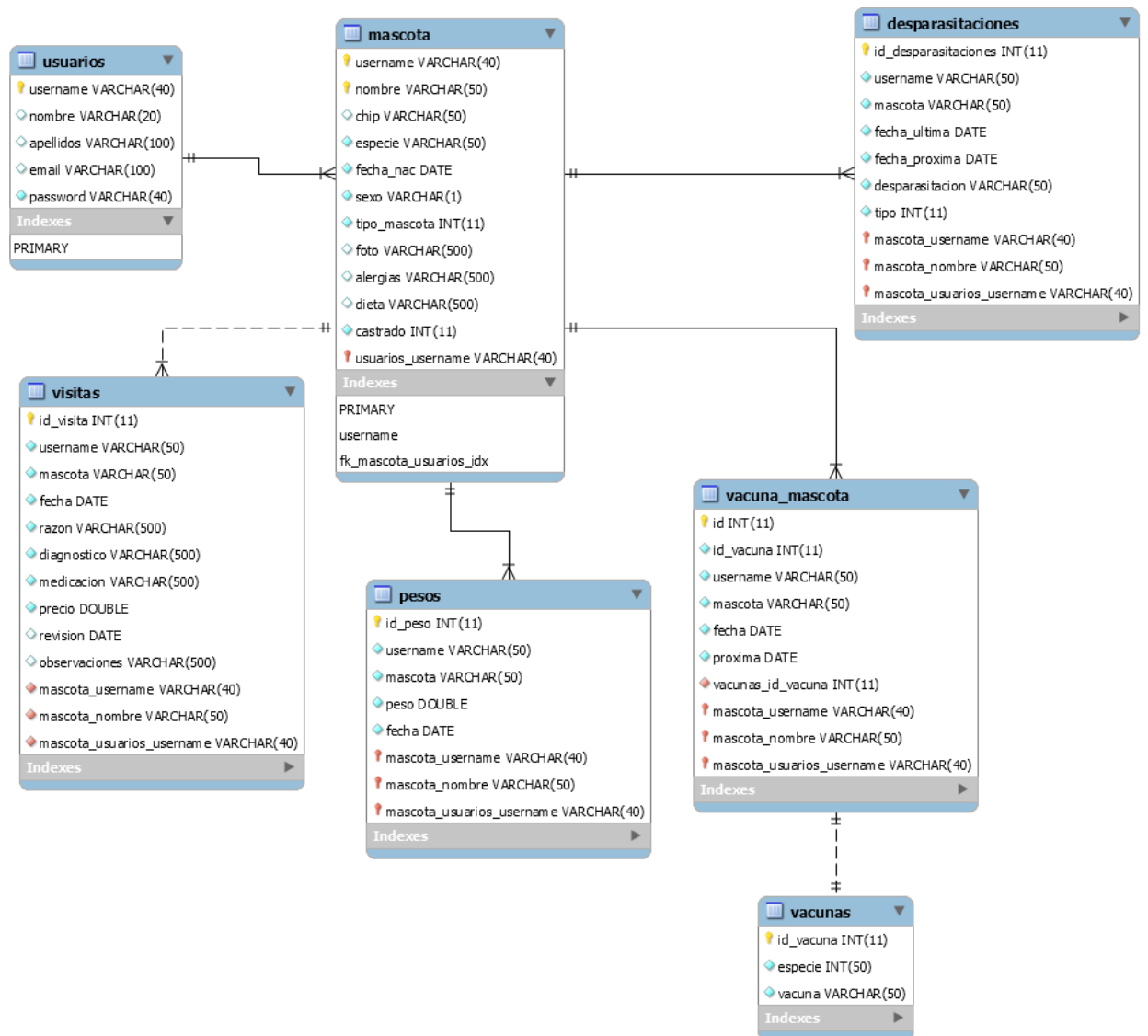


Figura 6-2: Diagrama de implementación de la base de datos

6.5 Interfaz de usuario

La interfaz de usuario ha sido desarrollada utilizando los ficheros XML proporcionados por el IDE de Android Studio. La interfaz ha sido diseñada buscando la mayor sencillez posible.

6.5.1 Registro del usuario

Como se muestra en la figura 6-3, en esta pantalla el usuario debe registrar sus datos personales, así como el usuario y la contraseña que usara en el futuro para poder acceder a la aplicación.



OurPets

Our
Pets

Usuario: ivan123alc

Contraseña:

Repite contraseña:

Nombre: Ivan

Apellidos: Zorita

Email: note@gmail.com

CANCELAR REGISTRAR

Figura 6-3: Registro en Ourpets

6.5.2 Login del usuario

Como se muestra en la figura 6-4, desde esta pantalla se accederá a la lógica de la aplicación. El usuario debe introducir el usuario y contraseña que usó para registrarse. Opcionalmente puede marcar que se recuerde su usuario y contraseña, con lo que no deberá introducirlos en el futuro.



OurPets

Our
Pets

Usuario:

Contraseña:

☐ Recordar usuario ☐ Recordar contraseña

REGISTRAR LOGIN

Figura 6-4: Login en Ourpets

6.5.3 Listado y alta de mascotas

La siguiente pantalla muestra el listado de las mascotas que ha dado de alta el usuario. Como se muestra en la figura 6-5 también se dispone de un botón de alta de mascotas.

Como se muestra en la figura 6-6, para dar de alta a una mascota se debe rellenar un formulario con los datos, así como el chip para los gatos y los perros. Opcionalmente se puede añadir una foto de la mascota, si no se añade aparecerá la foto por defecto para la especie concreta.

The figure consists of two side-by-side screenshots of a mobile application interface for managing pets, titled 'OurPets'.

The left screenshot (Figura 6-5) displays a list of pets. A yellow folder icon is shown with a small photo of a husky and the text 'Yako Perro'. Below the list is a green circular button with a white plus sign, indicating an option to add a new pet.

The right screenshot (Figura 6-6) shows the registration form for a pet. It includes a photo of a husky and three icons for adding a photo, taking a photo, or deleting a photo. The form fields are as follows:

- Nombre:** Yako
- Fecha Nacimiento:** 06-05-2016
- Especie:** Perro
- Sexo:** ☐ F ☒ M
- Castrado:** ☐ Si ☒ No
- Alergias:** No
- Dieta:** No
- Chip:** hskdjkf

At the bottom right of the form are two green buttons: 'CANCELAR' and 'GUARDAR'.

Figura 6-5 y 6-6: Listado y alta de mascotas

6.5.4 Detalles de la mascota

Una vez se añade una nueva mascota o hacemos clic en una de las ya registradas, pasaremos a la pantalla de detalles de la mascota.

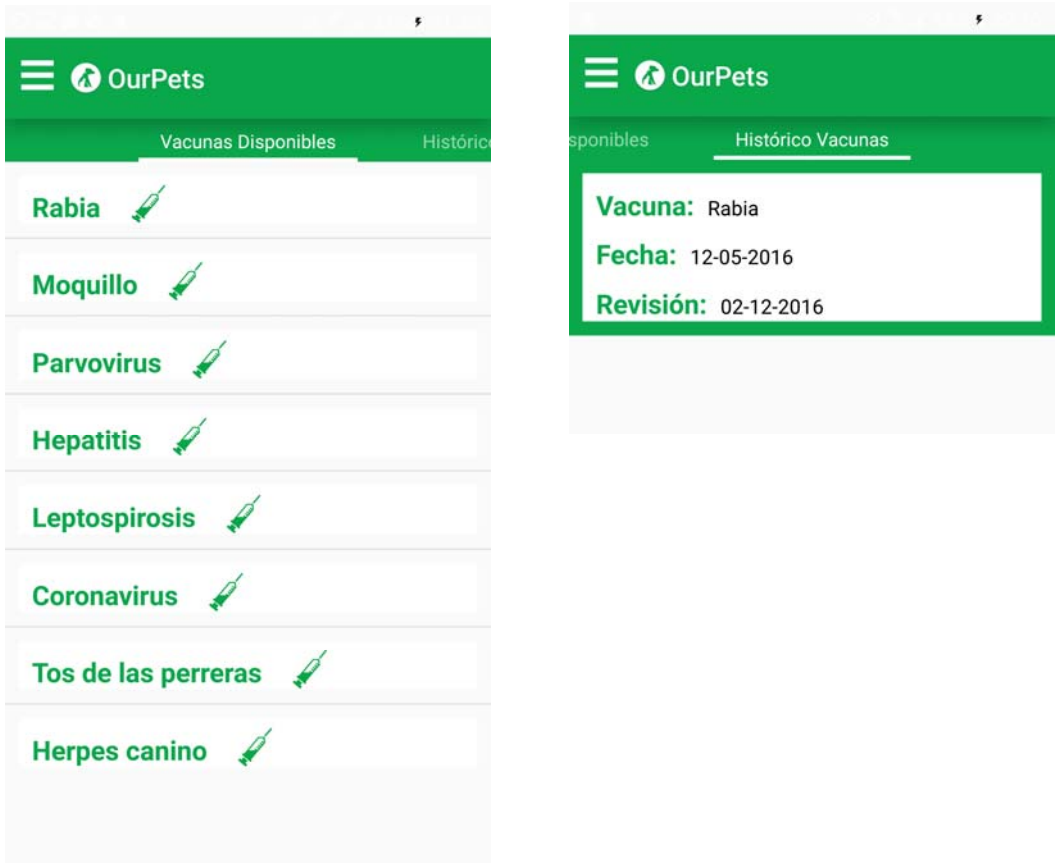
Como se muestra en la figura 6-7, esta pantalla nos muestra la información de la mascota, y desde aquí podremos navegar a las pantallas de desparasitación, vacunas, etc.



Figura 6-7: Detalles mascota

6.5.5 Vacunas

Esta pantalla muestra dos fragmentos, como se muestra en las figuras 6-8 y 6-9. La primera pantalla muestra un listado de vacunas que se pueden añadir a la mascota, para añadirlas habrá que tocar en la vacuna disponible. La segunda muestra el histórico de vacunaciones de la mascota.



Figuras 6-8 y 6-9: Vacunas disponibles e histórico de vacunas

6.5.6 Alta e histórico de desparasitaciones

Como se muestra en la figura 6-10, esta pantalla muestra el histórico de desparasitaciones de la mascota, con sus detalles. En esta pantalla se dispone de un botón para añadir nuevas desparasitaciones.

Para añadir una nueva desparasitación se debe rellenar un formulario con los detalles de la desparasitación, así como una fecha con la próxima desparasitación, lo que generará una notificación cuando llegue dicha fecha.

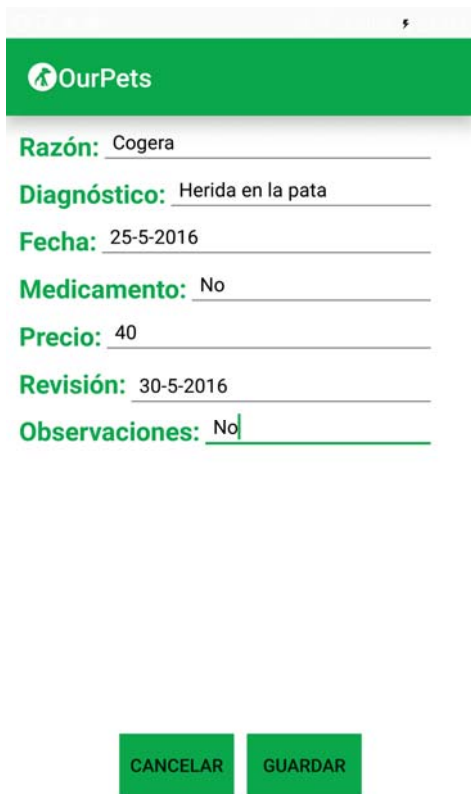


Figura 6-10: Listado de desparasitaciones

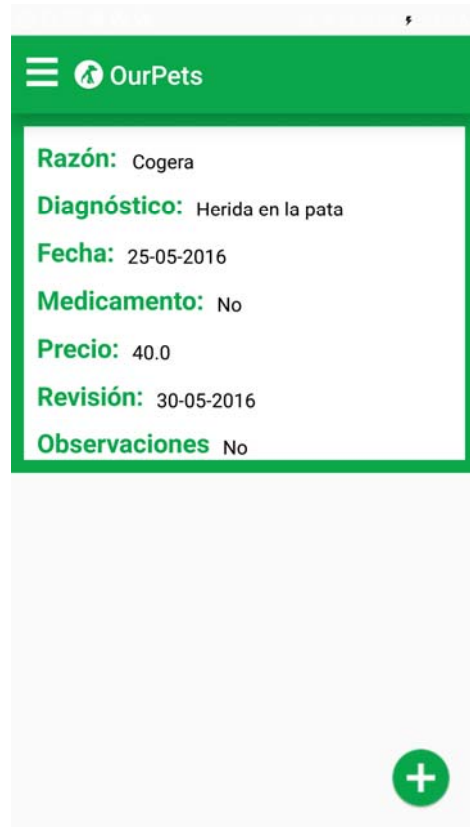
6.5.7 Alta e histórico de consultas al veterinario

Esta pantalla, como se ve en la figura 6-12, muestra el histórico de visitas al veterinario de la mascota. Tiene un botón para poder añadir nuevas visitas.

Cuando clicamos el botón de añadir, se mostrará un formulario. Se deberán rellenar los detalles de la nueva visita además del campo próxima revisión que genera una notificación el día indicado.



The screenshot shows the 'OurPets' app interface. At the top is a green header with a white icon of a person and a dog, and the text 'OurPets'. Below the header is a form with the following fields: 'Razón: Cogera', 'Diagnóstico: Herida en la pata', 'Fecha: 25-5-2016', 'Medicamento: No', 'Precio: 40', 'Revisión: 30-5-2016', and 'Observaciones: No'. At the bottom of the form are two green buttons: 'CANCELAR' and 'GUARDAR'.



The screenshot shows the 'OurPets' app interface. At the top is a green header with a white icon of a person and a dog, and the text 'OurPets'. Below the header is a list of consultations. Each consultation is represented by a green box containing the following information: 'Razón: Cogera', 'Diagnóstico: Herida en la pata', 'Fecha: 25-05-2016', 'Medicamento: No', 'Precio: 40.0', 'Revisión: 30-05-2016', and 'Observaciones: No'. At the bottom right of the screen is a green circular button with a white plus sign.

Figuras 6-11 y 6-12: Alta e histórico de consultas

6.5.8 Alta y gráfica de pesos

Ourpets ofrece la posibilidad de llevar un histórico del peso de la mascota. Esto se realiza desde la misma pantalla, pero en dos fragmentos diferentes, como se muestra en las figuras 6-13 y 6-14.

En la primera pantalla se podrá registrar el peso de la mascota, que se mostrará en una tabla justo debajo del formulario. Si deslizamos hacia la derecha se encuentra la gráfica de pesos de la mascota, que muestra de forma lineal la evolución del peso de la mascota durante los distintos pesajes.

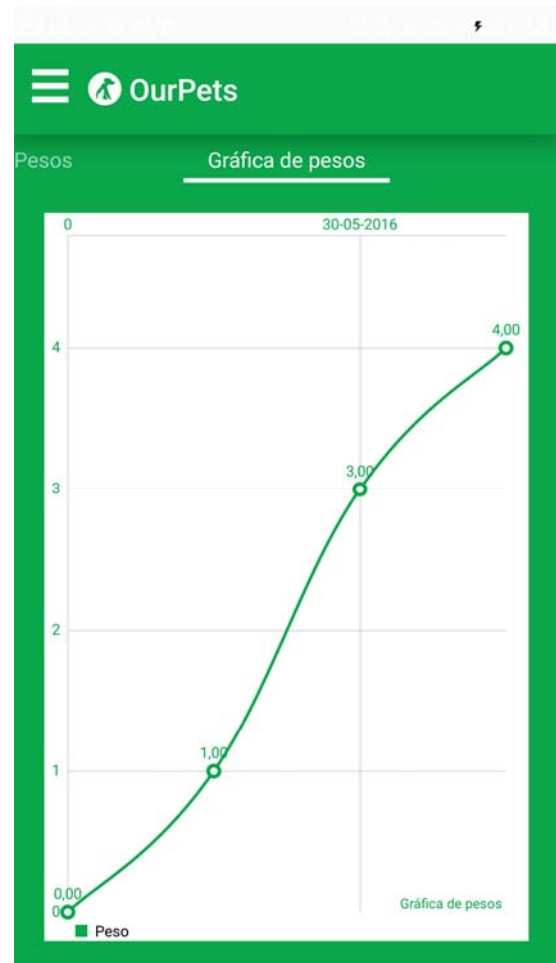
Pesos

Peso: 4

Fecha: 2-6-2016

+

Peso	Fecha
3.0	30-05-2016
1.0	21-05-2016



Figuras 6-13 y 6-14 Alta y gráfica de pesos

6.5.9 Veterinarios más cercanos

Como se muestra en la figura 6-15, esta pantalla muestra un listado, por orden de más cercano a más lejano, de los veterinarios de la zona en la que se encuentra el usuario. Además, si se hace click en uno de ellos nos redirigirá a Google Maps donde podremos ver cómo llegar.



Figura 6-15: Veterinarios cercanos

6.5.10 Ayuda

Esta pantalla será distinta según en qué sección de la aplicación nos encontremos. En esta pantalla se muestra información de ayuda acerca de cómo usar la aplicación en la sección en que nos encontramos.

6.5.11 Navegación

Para la navegación de la aplicación se ha usado un NavDrawer. Esto como se muestra en la aplicación, permite tener un menú desplegable personalizado con las opciones de navegación de la aplicación. Además, como se muestra en la figura 6-16, la sección actual en que nos encontramos aparecerá resaltada en verde.

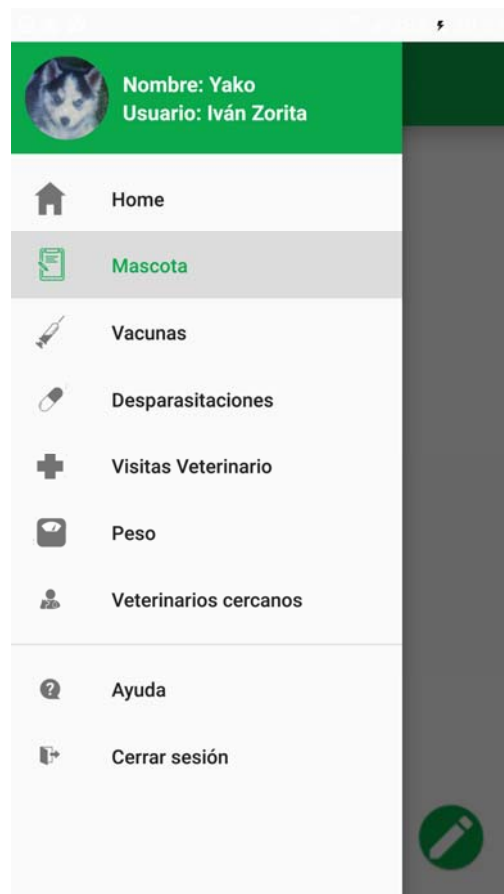


Figura 6-16: Navegación Ourpets

7 Pruebas

Aplicación, se va a comprobar que los distintos módulos de la aplicación funcionan correctamente, tanto de forma individual como en conjunto, para reducir el riesgo de errores en el uso de la aplicación cuando ésta se encuentre disponible al público.

7.1 Estrategia de pruebas

Antes de poder liberar la aplicación al cliente se debe probar la aplicación y asegurarse de que el riesgo de errores es mínimo o nulo. Para ello se debe elaborar un plan de pruebas que debe superar la aplicación. Como se ha comentado anteriormente la aplicación se subdivide en módulos. Por este motivo se probará la aplicación desde el módulo más pequeño, integrando poco a poco los módulos hasta realizar una prueba final con la aplicación completa.

7.1.1 Pruebas unitarias

Las pruebas unitarias se han realizado a cada módulo por separado para probar que cumplen con la funcionalidad con la que se implementaron.

Los test unitarios se han creado a medida que se han implementado los distintos módulos, de tal forma que se puedan probar todos y cada uno de los métodos de cada módulo.

Para realizar las pruebas unitarias en cuanto al código de la aplicación se ha utilizado Android Studio y Gradle. Gradle es el compilador de Java utilizado por Android Studio, desde el que podemos habilitar las pruebas unitarias con JUnit.

Para realizar las pruebas del servidor se ha hecho uso de la herramienta Postman, que ofrece la posibilidad de realizar peticiones Get y Post a las URL del servidor y poder detectar posibles errores de código.

7.1.2 Pruebas de integración

Estas pruebas se diseñan para comprobar la integración de los módulos más pequeños, formando poco a poco la funcionalidad completa de la aplicación.

Ourpets hace uso de la información almacenada en la base de datos por lo que se debe comprobar primero que la integración del servidor con la base de datos es correcta.

Una vez que se ha comprobado la integración de la base de datos con el servidor, se debe comprobar la integración del servidor con la aplicación.

7.1.3 Pruebas de interfaz

Estas pruebas consisten en comprobar si la interfaz de la aplicación se ve correctamente en dispositivos de todas las resoluciones disponibles.

Para realizarlas se ha ejecutado la aplicación en diferentes dispositivos de diferentes tamaños y resoluciones y se ha comprobado que no había una sobrecarga de campos en la pantalla, ni campos que se desplacen por falta de espacio, etc.

7.2 Resultados de las pruebas

En esta sección se van a describir los resultados obtenidos durante las distintas pruebas realizadas a la aplicación.

7.2.1 Pruebas unitarias

Durante las pruebas unitarias realizadas al código de la aplicación, se han detectado algunos problemas con el paso entre la actividad del diálogo de las fechas, al campo de texto que muestra la fecha, así como a la hora de quitarle el foco a dicho. También se detectaron problemas a la hora de obtener la imagen capturada con la cámara de fotos.

Durante las pruebas unitarias del servidor se han detectado algunos errores de sintaxis en las funciones que llaman a MySQL.

7.2.2 Pruebas de integración

Durante la integración del servidor con la base de datos se han observado múltiples errores de referencia a los campos de las tablas de la base de datos. También se encontraron errores en el fichero de conexión a la base de datos, que está incluido en todos los ficheros PHP.

Al realizar la integración del servidor con la aplicación, se detectó un problema al transformar las fechas del formato de base de datos. También se detectó un error en la descarga de las imágenes provenientes del servidor y su almacenamiento en la caché del Smartphone.

7.2.3 Pruebas de interfaz

En cuanto a la interfaz, se detectó que la etiqueta de la contraseña en la pantalla del login se descolocaba en dispositivos de tamaño reducido, por lo que se le redujo el tamaño de la fuente.

También se detectó que la imagen que aparecía en las notificaciones no se correspondía con la del logo de la aplicación.

8 Conclusiones y trabajo futuro

8.1 Conclusiones

Durante este trabajo de fin de grado se ha desarrollado una aplicación móvil nativa de Android, realizando ingeniería del software para llevarla a cabo. En su definición se eligió un desarrollo en cascada iterativo, por lo que desde el estudio del arte hasta la validación se han realizado varias iteraciones, solucionando errores o implementando pequeños cambios.

Durante el estudio del estado del arte se detectó escasez en este tipo de aplicaciones y para poder hacer un uso completo de sus funcionalidades en todo caso era necesario pagar de alguna forma u otra. El objetivo de Ourpets es mejorar las funcionalidades ofrecidas por las otras aplicaciones y hacerlo de forma gratuita.

El desarrollo de este trabajo de fin de grado, ha proporcionado al estudiante conocimientos avanzados sobre el desarrollo de aplicaciones nativas de Android, además de cómo gestionar un servidor web como AWS. También le ha proporcionado conocimientos sobre la gestión completa de proyectos.

8.2 Trabajo futuro

Después de la finalización del trabajo hemos encontrado dos líneas de trabajo para, en el futuro, poder mejorar la aplicación.

La primera sería desarrollar otro tipo de usuario, que sería el veterinario, de tal forma que pudiesen anunciar sus servicios desde la aplicación al usuario y después comunicarse entre ellos, proporcionando información útil para cualquier dueño de una mascota. Además, implementar un sistema de petición de citas desde la propia aplicación para que se pudiese solicitar una cita a un veterinario registrado en la aplicación sin necesidad de ir hasta el local para pedirla.

Otra línea de trabajo sería el desarrollo de un localizador GPS para la mascota. Esto permitiría al usuario poder encontrar su mascota en caso de perderla o que se escape de casa.

También se seguirá trabajando en la mejora de la funcionalidad implementada ya, a partir de la experiencia y el feedback generado por los usuarios finales que hagan uso de la aplicación.

Referencias

- [1] We Are Pets, “WAP Diary Beta”, Disponible:
<https://play.google.com/store/apps/details?id=app.wap&hl=es>
- [2] Bruno Diego Martins, “Ficha Animal VET”, Disponible:
<https://play.google.com/store/apps/details?id=com.brunodiego.animalrecords&hl=es>
- [3] Quality Compusoft S.L., “Mi veterinario”, Disponible:
<https://play.google.com/store/apps/details?id=es.winvet.mivet&hl=es>
- [4] 11 Pets, “11pets: Cuida tu mascota”, Disponible:
<https://play.google.com/store/apps/details?id=com.m11pets.elevenpets&hl=es>
- [5] Luca Biasotto, “Dog Health”, Disponible:
<https://play.google.com/store/apps/details?id=it.lobofun.doghealt&hl=es>
- [6] IntelliJ, “Android Studio”, Disponible:
<https://developer.android.com/studio/index.html>
- [7] Oracle, “MySQL”, Sitio oficial: <https://www.mysql.com/>
- [8] Oracle, “MySQL Workbench”, Disponible:
<http://dev.mysql.com/downloads/workbench/>
- [9] Oracle, “Java”, Disponible: <https://www.java.com/es/download/>
- [10] Amazon, “Amazon Web Services”, Sitio oficial: <https://aws.amazon.com/es/>

Glosario

API: Application Programming Interface.

SQL: Structured Query Language.

AWS: Amazon Web Service.

XML: eXtensible Markup Language.

JSON: JavaScript Object Notation.

IDE: Integrated Development Environment.

